

Appl. No. : 09/808,612
Filed : March 14, 2001

REMARKS

Claims 1-35 were pending in the application. By this paper, Claims 1, 14, 15, 19-24, 33 and 35 have been amended and Claims 27 and 28 have been cancelled without prejudice. Accordingly, Claims 1-26 and 29-35 are presented herein for examination.

5

Request for Continued Examination (RCE)

Applicant files herewith an RCE for continued prosecution of the above-referenced application. Applicant notes that per the PTOL Form 303 provided with the Office Action, Applicant's prior amendments were not entered (See Pars. 3 and 7.). Accordingly, Applicant
10 presents herein amendments and remarks identical to those previously provided in its October 3, 2005 response, based on the foregoing non-entry of this prior amendment.

§102 Rejections

Claim 1 – Per pages 2-3, 8-9 of the Office Action, Claim 1 stands rejected as being
15 anticipated by Merks, et al. (US Pat. No. 6,516,460, hereinafter “Merks”). Specifically, the Examiner contends there is nothing in the “claimed limitations of independent claim 1 to distinguish the claimed invention over the polling process taught by Merks.” Applicant respectfully disagrees. Claim 1 as presented herein sets forth, *inter alia*, switching without determining whether an event of interest has occurred. Applicant again notes that Merks teaches
20 a “waiting” step as part of its polling or switching process.

Applicant has herein amended Claim 1 to further clarify what applicant regards as his invention; specifically applicant has amended Claim 1 to include, *inter alia*, switching without determining whether a debug event of interest has occurred. Support for this limitation is provided at, *inter alia*, page 8, lines 9-22 of the specification as filed.

25 Merks, on the other hand, must wait to determine whether a debug event has occurred prior to switching (See e.g., Col. 7, lines 59-62). Therefore, not only does Merks not anticipate each and every limitation of Claim 1, Merks actually teaches away from the functionality of Claim 1 as currently amended, in that Merks must determine whether a debug event has occurred prior to switching. Applicant submits that Claim 1 as amended herein overcomes the Examiner's
30 § 102(e) rejection, and accordingly places Claim 1 in condition for allowance.

Appl. No. : 09/808,612
Filed : March 14, 2001

Claim 23 – Per page 8 of the Office Action, Claim 23 stands rejected as being anticipated by Merks. Applicant has herein amended Claim 23 to include the limitation of “obtaining status information from each of said processors based at least in part on said polling interval and outputting said status information to a user perceivable medium, said act of obtaining occurring without determining whether a debug event has occurred.” Support for Applicant’s amendment is replete throughout the specification, including e.g., page 8, lines 9-22 and page 14, lines 8-9. Merks does not teach nor suggest such functionality. Rather, Merks teaches checking for any waiting processes, and only after detecting that a debug event has occurred during a time out period does Merks obtain status information to output to a graphical user interface. Merks provides in relevant part:

“Should a debug event occur, the debug event is processed 170 by, for example, updating the process information in the debugger by, for example, refreshing the state and views of the user interface of the debugger, reporting the debug event, creating a new view in the debugger's user interface in the case of the spawning of a child process, or removing a view in the debugger's user interface in the case of the termination or completion of a process (not shown).” Col.8, lines 37-44.

Further, not only does Merks not anticipate each and every limitation of Claim 23, Merks teaches away from the invention of amended Claim 23 as Merks requires a debug event to occur before outputting status information to a user perceivable medium. Applicant submits that amended Claim 23 (and Claims 29 and 30, which either depend directly or indirectly therefrom) are therefore in condition for allowance.

§103 Rejections

Per page 5 of the Office Action, the Examiner continues to contend that the motivation to combine Merks with Davis et al. (US Pat. No. 6,230,307, hereinafter “Davis”) is proper in support of his rejection of Applicant’s Claims under 35 U.S.C. §103. Applicant respectfully again disagrees with the Examiner’s assertion.

Specifically, the Examiner cites In Re Rouffet as supporting, *inter alia*, the proposition that a skilled artisan would have made an effort to become aware of what capabilities had been

Appl. No. : 09/808,612
Filed : March 14, 2001

developed in the market place, and hence would have knowingly modified Merks with the teachings of Davis. While applicant concedes that knowledge of persons of ordinary skill in the art may be used as a possible source for finding motivation to combine references, the Examiner must still find an objective reason (whether express or implied) to combine the references. See e.g., MPEP 2143.01 reproduced in relevant part below:

“A statement that modifications of the prior art to meet the claimed invention would have been “‘well within the ordinary skill of the art at the time the claimed invention was made’ ” because the references relied upon teach that all aspects of the claimed invention were individually known in the art is not sufficient to establish a prima facie case of obviousness without some objective reason to combine the teachings of the references. *Ex parte Levengood*, 28 USPQ2d 1300 (Bd. Pat. App. & Inter. 1993). See also *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1318 (Fed. Cir. 2000)” {Emphasis added}

Respectfully, the Examiner’s view that “a skilled artisan working in a competitive environment would have made an effort to become aware of what capabilities had already been developed in the market place, and hence would have been aware of, and known to seek out the relative teachings of the problem to be solved” amounts, in Applicant’s view, to nothing more than subjective and impermissible hindsight and does not support an objective conclusion that there was any motivation to combine.

Further, per page 6 of the Office Action, the Examiner offers as support for the motivation to combine the Merks and Davis references that Merks teaches, *inter alia*, “solving the problem of debugging processes in a distributed (multi) processor environment.” Applicant respectfully can find no support that Merks even remotely suggests solving the problem of debugging processes in a distributed multi-processor environment. Applicant can only surmise that the Examiner is equating the term “process” with “processor”, which in Applicant’s view is clearly improper since the multiple processes need not by any means be on multiple processors. The abstract of Merks sets forth in relevant part:

“Methods; systems and articles of manufacture comprising a computer usable medium having computer readable program code means therein are provided for debugging multiple related processes simultaneously...”

Applicant therefore reasserts that the Examiner's reasoning for the combination of Merks and Davis is inadequate in light of the MPEP and United States patent law. However, Applicant submits that the objectionable combination of Merks and Davis is moot in light of the following amendments and arguments.

5
Claim 1 – With regards to the §103 rejections of Claim 1, the Examiner is directed to Applicant's response with respect to the §102 rejections discussed above. Applicant believes that not only is every limitation of Claim 1 as amended not taught in either Merks nor Davis, Merks actually teaches away from Claim 1, as Merks must detect a debug event before switching.

10
Claim 14 – Per page 14 of the Office Action, Claim 14 stands rejected under §103 as being unpatentable over Merks in view of Davis. Applicant traverses the Examiner's rejection; specifically, Applicant believes that neither Merks nor Davis actually teach each and every limitation of Claim 14. Claim 14 sets forth, *inter alia*, a system for debugging heterogeneous
15 processors while Merks is primarily concerned with “debugging multiple related processes simultaneously”. Applicant believes that not only does Merks not suggest the concept of debugging heterogeneous processors; Merks is further inconsistent with the concept of debugging heterogeneous processors, and teaches away from the invention of Claim 14 by its stated purpose of “debugging multiple related processes”.

20 Applicant also reiterates its belief that the Examiner's contention that Merks somehow teaches or suggests debugging in a distributed multi-processor environment does not appear, from Applicant's reading, to be supported anywhere in the Merks specification.

Further, Applicant has herein amended Claim 14 to include limitations relating to the heterogeneous processors further including at least two differing instruction sets. Support for this
25 amendment is found, *inter alia*, at page 13, lines 16-21 of Applicant's specification. While applicant believes that Claim 14 had previously distinguished over both Merks and Davis, Applicant believes that these added amendments yet further distinguish Applicant's invention over the prior art. Neither Merks nor Davis teach or suggest such functionality as contained in amended Claim 14. Applicant therefore submits that Claim 14 and all claims depending either
30 directly or indirectly therefrom are in condition for allowance.

Claim 19 - By this paper, Applicant has amended Claim 19 to include limitations relating to the recited act of continuously switching comprising switching between said processes irrespective of the status of a debug event. Applicant notes that in contrast, Merks teaches using a “waiting” step 150 and check for a debug event as part of its polling or switching process. The approach of Merks thereby necessarily induces comparatively lengthy “wait” periods and checks for debug events; see, e.g., Col. 8, lines 10-19 of Merks:

“The “waiting” step 150 is limited by a timeout period to check for a debug event. In a Windows NT embodiment, the debugger would check a debug event queue, by means of the WaitForDebugEvent() command, for the timeout period to see if a debug event has occurred. If a debug event had occurred during the execution of that command or occurred before initiation of that command, the command would return an indication of the occurrence of a debug event because of the presence of a debug event of the queue. Otherwise, there would be no indication of a debug event.” {Emphasis added}

Applicant’s invention as set forth in amended Claim 19 advantageously obviates any such debug event check, thereby greatly expediting the debug process. Applicant submits that these amendments overcome the Examiner’s §103 rejections of Claim 19.

Claim 20 – Per pages 14-15 of the Office Action, Claim 20 stands rejected under §103 as being unpatentable over Merks in view of Davis. Applicant has herein amended Claim 20 to include limitations relating to debugging a plurality of distributed programs across heterogeneous processors. Further, the act of selectively permitting at least a portion of said processes to operate further includes simulating the failure of at least one of said heterogeneous processors using a single one of said at least one interface. Support for this amendment can be found, *inter alia*, at page 7, lines 1-5 of Applicant’s specification. Applicant submits that neither Merks nor Davis teach or suggest such functionality as contained in amended Claim 20 presented herein, and hence Claim 20 is in condition for allowance.

Claim 21 – Per page 14 of the Office Action, Claim 21 stands rejected under §103 as being unpatentable over Merks in view of Davis. Applicant has herein amended Claim 21 to

Appl. No. : 09/808,612
Filed : March 14, 2001

include, *inter alia*, limitations relating to optimizing the operation of a multi-processor system comprising a plurality of heterogeneous software processes.

Applicant submits that Claim 21 as amended can not be rendered obvious by Merks in combination with Davis as a matter of law. Merks teaches “computer readable program code means therein are provided for debugging multiple related processes simultaneously. Merks further goes on to define a related process as “processes that are spawned, directly or indirectly, from a single process.” See Col. 1, lines 25-27 of Merks. Merks therefore teaches away from a method of optimizing the operation of a multi-processor system comprising heterogeneous software processes, as it is only concerned with processes that are spawned, directly or indirectly, from a single as opposed to heterogeneous process.

Claim 22 – Per page 14 of the Office Action, Claim 22 stands rejected under §103 as being unpatentable over Merks in view of Davis. Applicant has herein amended Claim 22 to include limitations relating to coordinating the operation of at least one simulation process with hardware processes to also include at least two heterogeneous hardware processes. These two heterogeneous hardware processes also further comprise at least two different instruction set architectures. Support for these amendments can be found *inter alia* at page 13 of Applicant’s specification.

Applicant submits that neither Merks nor Davis teach or suggest such functionality. Therefore, Applicant respectfully submits that Claim 22 as amended is in condition for allowance.

Claim 23 – See discussion of Claim 23 above with respect to the Examiner’s §102 rejections; similar reasoning applies with respect to the Examiner’s §103 rejections of Claim 23.

Claim 24 – Per pages 14-15 of the Office Action, Claim 24 stands rejected under §103 as being unpatentable over Merks in view of Davis. Applicant respectfully disagrees with the Examiner’s contention that both Davis and Merks disclose simulation (emulation) debugging in a distributed multi-processor environment. Specifically, Applicant can find no support that Merks ever contemplates; *inter alia*, a distributed multi-processor environment.

Applicant has however amended Claim 24 to include limitations relating to the gathering of status information that occurs on a dynamic per-process time interval, thereby rendering the Examiner's rejection moot. Support for this amendment can be found, *inter alia*, at Page 9, lines 3-15 of Applicant's specification as filed. Applicant believes that neither Merks nor Davis teach nor remotely suggest such dynamic per-process time intervals, and hence Claim 24 is independently allowable as currently amended as well. In addition, as Claims 25 and 26 depend either directly or indirectly from Claim 24, Applicant believes these claims are in condition for allowance as well.

Claim 31 – Per page 16 of the Office Action, Claim 31 was rejected under §103 as being unpatentable over Merks in view of Davis and further in view of Katzeff (US Pat. No. 5,101,491; hereinafter “Katzeff”). Applicant respectfully submits that the combination of Merks, Davis and Katzeff is improper. Applicant believes that Merks actually teaches away from the invention of Claim 31. Specifically Applicant notes that Merks teaches away from “switching on a periodic basis between individual ones of said processes to obtain status information relating thereto irrespective of the status of a debug event associated with one of said processes” as recited in Claim 31. Recall that as previously discussed herein, Merks requires a “waiting” step to check to see whether a debug process has occurred prior to processing the debug event (See e.g., Fig. 1). Therefore, applicant believes that not only does Merks not teach each and every element of Claim 31, Merks actually teaches away from the invention of Claim 31 by requiring the detection of a debug event before processing the debug event. Such waiting step could hardly be characterized as falling under the rubric of “irrespective of the status.”

Furthermore, as Claim 32 depends directly from Claim 31, Applicant believes Claim 32 is in condition for allowance as well.

Claim 33 – Per page 15 of the Office Action, Claim 33 stands rejected under §103 as being unpatentable over Merks in view of Davis and further in view of Katzeff. Applicant has herein amended Claim 33 to include limitations relating to debugging a plurality of processes on a plurality of heterogeneous processors using a debugger program, to include a process step wherein the plurality of processes are adapted to gather status information regarding respective

Appl. No. : 09/808,612
Filed : March 14, 2001

ones of said heterogeneous processors as controlled by the debug process, the gathering of status information occurring on a dynamic per-process time interval. Support for Applicant's amendment can be found, *inter alia*, at page 9, lines 3-15 of Applicant's specification. Applicant submits that neither Merks nor Davis nor Katzeff teach or suggest such functionality.

5 Applicant further submits that Claim 33 and Claim 34, which depends directly from amended Claim 33, are both novel and non-obvious in light of the prior art, and therefore in condition for allowance.

10 **Claim 35** – Per page 16 of the Office Action, Claim 35 stands rejected under §103 as being unpatentable over Merks in view of Davis and in further view of Katzeff. Applicant respectfully disagrees with the Examiner's assertions; specifically, Applicant contends that the utilization of Merks is improper in combination with the other two cited references in light of the claimed subject matter. Claim 35 sets forth in relevant part that said acts of running and monitoring said heterogeneous processes are all performed using a single debugging program.
15 Applicant believes that Merks more correctly stands for the opposite proposition, namely using a debugging program on multiple related processes, and not heterogeneous processes.

Further, Claim 35 recites running said at least one hardware processor so as to (i) execute at substantially the speed with which execution would occur if it were not running under the control of a debugger, and (ii) monitor the status of said at least one hardware processor using a
20 single thread of control, said single thread not significantly slowing said execution. Merks on the other hand must implement a "waiting" step and check for a debug event as part of its overall functionality resulting in substantial slowing of its program execution. Applicant's invention as set forth in amended Claim 35 advantageously obviates any such debug event check, thereby greatly expediting the debug process, especially in the context of heterogeneous processes being
25 debugged, where time is a critical attribute.

Other Remarks

Applicant hereby specifically reserves the rights to prosecute claims of different or broader scope, including those cancelled herein, in a continuation or divisional application.

Appl. No. : 09/808,612
Filed : March 14, 2001

Applicant notes that any claim cancellations or additions made herein and not substantively discussed above are made solely for the purposes of more clearly and particularly describing and claiming the invention, and not for purposes of overcoming art. The Examiner should infer no (i) adoption of a position with respect to patentability, (ii) change in the Applicant's position with respect to any claim or subject matter of the invention, or (iii) acquiescence in any way to any position taken by the Examiner, based on such cancellations or additions.

Furthermore, any remarks made herein with respect to a given claim or amendment are intended only in the context of that specific claim or amendment, and should not be applied to other claims, amendments, or aspects of Applicant's invention.

If the Examiner has any questions or comments which may be resolved over the telephone, he is requested to call the undersigned at (858) 675-1670.

Respectfully submitted,

GAZDZINSKI & ASSOCIATES

Dated: November 29, 2005

By:



Robert F. Gazdzinski
Registration No. 39,990
11440 West Bernardo Court, Ste. 375
San Diego, CA 92127
(858) 675-1670 voice
(858) 675-1674 fax